



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Multilevel Monte Carlo simulation of Coulomb Collisions

M. S. Rosin, L. F. Ricketson, A. M. Dimits, R. E.  
Caflisch, B. I. Cohen

October 10, 2013

Journal of Computational Physics

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# Multilevel Monte Carlo simulation of Coulomb collisions

M.S. Rosin<sup>a,\*</sup>, L.F. Ricketson<sup>a</sup>, A.M. Dimits<sup>b</sup>, R. Caflisch<sup>a,c</sup>, B.I. Cohen<sup>c</sup>

<sup>a</sup>*Mathematics Department, University of California at Los Angeles, Los Angeles, CA 90036, USA*

<sup>b</sup>*Lawrence Livermore National Laboratory, L-637, P.O. Box 808, Livermore, CA 94511-0808, USA*

<sup>c</sup>*Institute for Pure and Applied Mathematics, University of California at Los Angeles, Los Angeles, CA 90095, USA*

---

## Abstract

We present a new, for plasma physics, highly efficient multilevel Monte Carlo numerical method for simulating Coulomb collisions. The method separates and optimally minimizes the finite-timestep and finite-sampling errors inherent in the Langevin representation of the Landau-Fokker-Planck equation. It does so by combining multiple solutions to the underlying equations with varying numbers of timesteps. For a desired level of accuracy  $\varepsilon$ , the computational cost of the method is  $O(\varepsilon^{-2})$  or  $O(\varepsilon^{-2}(\ln \varepsilon)^2)$ , depending on the underlying discretization, Milstein or Euler-Maruyama respectively. This is to be contrasted with a cost of  $O(\varepsilon^{-3})$  for direct simulation Monte Carlo or binary collision methods. We successfully demonstrate the method with a classic beam diffusion test case in 2D, making use of the Lévy area approximation for the correlated Milstein cross terms, and generating a computational saving of a factor of 100 for  $\varepsilon = 10^{-5}$ . We discuss the importance of the method for problems in which collisions constitute the computational rate limiting step, and its limitations.

*Keywords:* Coulomb collisions, plasma, Monte Carlo, multilevel Monte Carlo, particle in cell

---

## 1. Introduction

In many regimes of practical importance, Coulomb collisions are an integral part of any accurate plasma description. For highly collisional systems, they are essential for closing the moment hierarchy of the kinetic equation and deriving microphysical expressions for the fluid transport coefficients. For marginally collisional systems with order one Knudsen numbers, they play an important role in the dynamics, for example in tokamak edge plasmas [1, 2], inertial confinement fusion [3], and astrophysics [4]. For weakly collisional, or ‘collisionless’ systems, they regulate nonlinear phase space cascades of generalized energy and entropy [5, 6], and can be used to understand and control grid errors in numerical simulations.

This paper presents a new (for plasma physics applications) accurate and efficient multi-(time-) level computational method for collisional kinetic problems, and is especially useful for systems in the low Knudsen number, i.e. highly collisional, regime. The method leverages a stochastic differential equation (SDE), or Langevin, approach to solving the kinetic equation particle-wise. It then combines the solutions using the multilevel Monte Carlo (MLMC) scheme,

---

\*Corresponding author

Email address: msr35@math.ucla.edu (M.S. Rosin)

initially developed for applications in financial mathematics [7] and now used in a wide variety of disparate areas [8].

The MLMC method generates computational savings by separating and independently minimizing the finite-timestep and finite-sampling errors inherent in any numerical SDE solver. Analogous to deterministic multigrid methods [9], the method builds a solution calculated from a weighted sum, over different ‘levels’  $l$ , of successively refined building-block solutions obtained by direct methods like, for example, the Euler-Maruyama or Milstein discretizations. The so called ‘strong convergence’ properties of these direct schemes determine the efficiency of the MLMC scheme in terms of a *global* error bound  $\varepsilon$  in expectation, over all particles, of the time-integrated solution of the underlying SDE.

The solutions returned by the MLMC method are accurate approximations of the mean, with respect to the particle distribution function  $f$ , of any Lipschitz ‘payoff’ function  $P$  of the generalized phase space coordinates. This can include the physically important macroscopic velocity moments of  $f$ , such as the density, fluid velocity, and temperature, that are governed by the moments of the underlying kinetic equation. For example, in the case of a homogeneous, force-free, collisional plasma, the fluid velocity is governed by the first moment equation

$$n \frac{\partial u_i}{\partial t} = R_i,$$

where  $n = \int f d^3\mathbf{v}$ ,  $u_i = \int f v_i d^3\mathbf{v}$  and  $R_i = \int (\partial f / \partial t)_{\text{coll}} v_i d^3\mathbf{v}$  are the macroscopic density, fluid velocity and mean collisional transfer of momentum, respectively. Here  $t$  is time,  $\mathbf{v}$  is the particle velocity with components  $v_i$ , and  $(\partial f / \partial t)_{\text{coll}}$  is the Landau-Fokker-Planck collision operator [10]. Unlike other approaches based on solving derived fluid equations, the MLMC method does not rely on collisional closures or *ad hoc* truncation schemes. The macroscopic solutions accurately reflect the underlying microscopic dynamics because the kinetic equation is solved directly.

The advantages of the MLMC method should be considered within the broader context of numerical collision methods for kinetic problems: particle-based, hybrid, and continuum methods [11, 12]. Each has its own merits. Particle based methods are simple, direct and converge at a rate independent of the number of dimensions, but carry a stochastic error that depends on the number of simulation particles as  $O(N^{-1/2})$ . Hybrid methods are versatile and efficient, but only lead to efficiency gains for partially thermalized systems [13]. Continuum methods are deterministic, but scale poorly with increased total (velocity plus spatial) dimension and lack the robustness of particle methods. They must also respect stability and CFL-like constraints on their discretization - even in the absence of mean fields.

For Monte Carlo simulations (pure particle and particle-based hybrid methods), binary collisions, for example the methods of Takizuke and Abe, and Nanbu, are a popular option [14, 15]. These collision methods fall into a class of quasi-Maxwellian Boltzmann equations that have been shown to be no less accurate than  $O(\Delta t^{1/2})$  in terms of their global truncation error [16]. Related analytic and numerical studies confirm this lower bound, and these schemes have been argued to be as fast as  $O(\Delta t)$  in the best case scenario [17, 18, 19]. The sampling error of the methods, governed by the Central Limit Theorem, scales as  $O(N^{-1/2})$ . The Langevin-based or SDE description presents an alternative.

Existing computational Langevin collision models have largely focused on the lowest order ‘Euler-Maruyama’ approximation to the underlying Langevin equation. Starting with Ivanov and Shvets [20, 21] various collision models have been developed that evolve some subset of the particle’s energy, pitch angle and azimuthal angle [22, 23, 24, 25, 26, 27, 28]. In their most basic

forms, the ‘weak’ convergence errors associated with these schemes are, like optimal binary methods,  $O(\Delta t)$ . Some of the models also include advanced numerical techniques like grid-based schemes or schemes that use the Euler-Maruyama discretization as a building block in, for example, predictor-corrector schemes. Further extensions include self-consistent field models [29], gyrokinetic applications [30], and laser-plasma applications [31, 32].

Beyond the Euler-Maruyama scheme, the next approximation in the hierarchy of higher order schemes is the ‘Milstein’ scheme. Its basic weak convergence error is also, like the Euler-Maruyama scheme,  $O(\Delta t)$ , but its strong convergence properties are improved. In one dimension, the Milstein terms are easy to implement [27, 28]. In higher dimensions, two or more, the complex statistics and statistical correlations in orthogonal Milstein terms prevent a simple description. Because collisions are a fundamentally multi-dimensional process in velocity space, even in reduced frameworks like gyrokinetics, this has been a major impediment for the application of higher-order Langevin methods in plasma physics. However, recent work provides a simple, efficient approximation to the statistically correlated component of the orthogonal Milstein terms and a proof of concept demonstration of their use for Coulomb collisions [33].

Existing SDE collision models and, in the best case, binary collision models have the same order of accuracy  $O(\Delta t)$ . Both methods also have the same computational cost  $\sim O(\varepsilon^{-3})$ , which comes from the product of a factor  $\varepsilon^{-1}$  from the timestepping cost and a factor  $\varepsilon^{-2}$  from the sampling cost (a result we derive in section 3.1). This is to be contrasted with the cost of the MLMC method, which uses discretized SDEs as building blocks.

The Euler-Maruyama MLMC scheme is  $O((\ln \varepsilon)^2/\varepsilon)$  faster than both SDE and binary methods, for the same level of accuracy. The Milstein MLMC method is even faster, offering a relative saving of  $O(1/\varepsilon)$ , and is optimal amongst all discretizations [34]. This paper provides a proof and demonstration of these results.

The layout of this paper is as follows. In section 2 we introduce the Langevin representation of the Landau-Fokker-Planck collision operator, and its basic numerical representation. In section 3 we review the MLMC method of Giles that uses, as its building block, the basic numerical representation of the collision operator. In section 4 we present the results of the MLMC method as applied to a collisional relaxation problem. In section 5 we describe some limitations of the method and sketch some potential avenues for extending it. Finally, in section 6 we summarize and conclude.

## 2. Coulomb-Langevin equations

### 2.1. Formulation

The starting point for most plasma collision models is the Landau-Fokker-Planck operator [10]. This describes the effect of many small-angle collisions on the evolution of the phase-space test-particle distributions function  $f_a \equiv f_a(t, \mathbf{v})$  of the charged plasma species  $a$

$$\frac{\partial f_a}{\partial t} = \frac{\partial f_a}{\partial t} \Big|_{\text{coll}} \equiv -\frac{\partial}{\partial v_i} \left( \left( \frac{\partial h}{\partial v_i} \right) f_a \right) + \frac{1}{2} \frac{\partial^2}{\partial v_i \partial v_j} \left( \frac{\partial^2 g}{\partial v_i \partial v_j} f_a \right), \quad (1)$$

where  $t$  is time,  $\mathbf{v}$  is velocity with components  $v_i$  and repeated indices are summed over. The Rosenbluth potentials  $h, g$  [35] are given by

$$(\partial^2 / \partial v_k \partial v_k) h = -4\pi \sum_b \Gamma (1 + m_a/m_b) f_b, \quad (2)$$

$$(\partial^4 / \partial v_k \partial v_m \partial v_k \partial v_m) g = -8\pi \sum_b \Gamma f_b, \quad (3)$$

where  $\Gamma = 4\pi q_a^2 q_b^2 \Lambda / m_a^2$ , the sum is over the index  $b$  of the plasma field-particle species  $f_b$ , mass is  $m$ , charge is  $q$ , and  $\Lambda$  is the Coulomb logarithm.

An alternative representation of the integro-differential Coulomb collision operator (1)-(3) is a drag-diffusion SDE for the random variable  $\mathbf{v}$ , describing the same stochastic memoryless (Markov) process. Under the assumption of white-noise forcing, the SDE description can be shown to be equivalent to the Fokker-Planck or forward Kolmogorov representation (Chapter 9.3, [36]).

Recasting the distribution function  $f_a$  as a sum of delta-function particles, indexed by  $k$  (henceforth repressed)

$$f_a(t, \mathbf{v}) = \sum_k \delta(\mathbf{v} - \mathbf{v}^k(t)), \quad (4)$$

the particle velocities are governed by Newton's Second Law, which in the special case of (1), corresponds to the SDE [21, 36]

$$dv_i = F_i dt + D_{ij} dW_j. \quad (5)$$

Here the total force is the sum of a deterministic drag force with coefficient  $F_i$  and a stochastic diffusion force with coefficient  $D_{ij}$  and Wiener, or Brownian, process  $W_i(t)$  with a normal probability density and variance

$$\mathbb{E} [ [W_i(t_2) - W_i(t_1)]^2 ] = |t_2 - t_1|,$$

where  $\mathbb{E}$  is the expectation. The Brownian motions are independent for each particle and component of the velocity. See Table 1 for a summary of notation.

In Cartesian coordinates, and adopting an 'Ito interpretation' [37],  $F_i$  and  $D_{ij}$  are related to (1) by

$$F_i = (\partial / \partial v_i) h, \\ D_{ij} = [(\partial^2 / \partial v_i \partial v_j) g]^{1/2},$$

and when  $f_a$  is in equilibrium, i.e. a Maxwellian with the same temperature and flow velocity as  $f_b$ ,  $F_i$  and  $D_{ij}$  are also related to each other by the Einstein<sup>1</sup> relations [38]:

$$F_i f_a + \frac{\partial}{\partial v_j} (D_{ij} f_a) = 0. \quad (6)$$

---

<sup>1</sup>In the general sense, 'Einstein relations' express the balance of deterministic and diffusive fluxes in a Fokker-Planck type equation satisfied in equilibrium. The work of Einstein described the positional motion of particles suspended in a fluid (undergoing Brownian motion), so that the discussion there was of fluxes crossing notional boundaries in configuration space. In this paper, the relevant fluxes are across boundaries in velocity space.

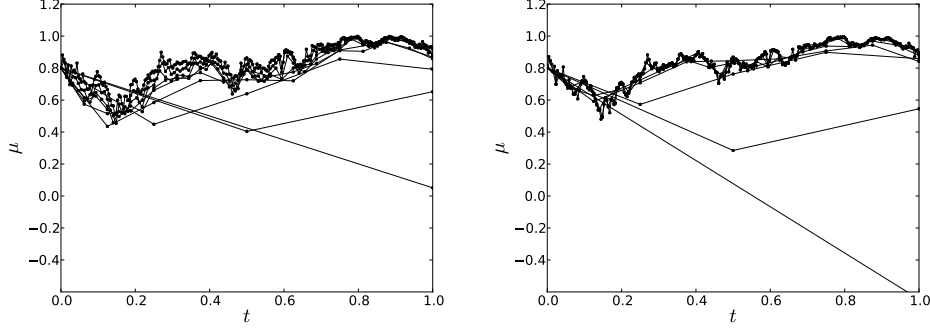


Figure 1: Collisional evolution of velocity-space coordinate  $\mu$  for a single particle using the Euler-Maruyama integration scheme (left) and Milstein scheme (right). Results are generated from (50) and (51) using successively compounded timesteps  $\Delta t_l = T2^{-l}$  for  $l = 0, 1, 2 \dots 8$ , and the same underlying Brownian path. The rapid convergence of the Milstein results, with increasing  $l$ , are indicative of the scheme's improved strong convergence properties relative to the Euler-Maruyama scheme. Pairs of paths with  $l, l-1$  are combined in the MLMC scheme to estimate  $\mathbb{E}[\mu]$ .

In curvilinear coordinate systems or for other stochastic calculuses, e.g. the ‘Stratonovich interpretation’,  $F_i$  and  $D_{ij}$  can appear as mixed coefficients of the drag and diffusion terms in (5) [37].

Without loss of generality, macroscopic forcing - electromagnetism, gravity, model terms - can be included in this formulation. Either directly in the coefficient of the deterministic drag term, or via an operator splitting procedure. Once it has been numerically discretized, (5) presents a simple method for including collisions or other stochastic processes in particle-in-cell codes. The method can also be applied to classes of stochastic kinetic systems more general than plasmas.

## 2.2. Numerical discretization

In general, solutions to (5) at time  $T$ ,  $\mathbf{v}(T)$  must be obtained numerically. Discretization can be achieved by an iterative (stochastic)-Taylor expansion in the finite timestep  $\Delta t_l = 2^{-l}T$ .

The simplest integration scheme is the Euler-Maruyama scheme, Fig. 1. It is

$$\Delta v_i = F_i \Delta t_l + D_{ij} \Delta W_j, \quad (7)$$

where  $\Delta v_i = v_i(t + \Delta t_l) - v_i(t)$ ,  $\Delta W_j = W_j(t + \Delta t_l) - W_j(t)$ , and under the Ito interpretation, the coefficients  $F_i, D_{ij}$ , and their derivatives are to be evaluated at time  $t$ . Solutions to (5) obtained using schemes like (7) and its higher order extensions, are said to be obtained *directly*, or using *single level* estimates, i.e.  $l = \text{constant}$ .

The weak and strong convergence properties of a direct scheme, like (7), can be defined in terms of its weak and strong errors [37, 33]. When solving for  $\mathbf{v}(T)$  these are, respectively, given by:

$$\begin{aligned} \varepsilon_W(\mathbf{v}, T, \Delta t) &= |\mathbb{E}[\mathbf{v}(T)] - \mathbb{E}[\mathbf{v}_l(T)]|, \\ \varepsilon_S(\mathbf{v}, T, \Delta t) &= \mathbb{E} \left[ |\mathbf{v}(T) - \mathbf{v}_l(T)|^2 \right]^{1/2}, \end{aligned}$$

where  $\mathbf{v}_l$  is the solution to (5) obtained using the finite timestep  $\Delta t_l$ . When solving for some Lipschitz function of  $\mathbf{v}(T), P[\mathbf{v}(T)]$ , the definition of the weak error (although not the strong error) must be generalized, so that [37]:

$$\varepsilon_W(P(\mathbf{v}), T, \Delta t) = |\mathbb{E}[P(\mathbf{v}(T))] - \mathbb{E}[P(\mathbf{v}_l(T))]|, \quad (8)$$

$$\varepsilon_S(P(\mathbf{v}), T, \Delta t) = \mathbb{E} \left[ |\mathbf{v}(T) - \mathbf{v}_l(T)|^2 \right]^{1/2}. \quad (9)$$

For single-valued initial conditions for the random variables, the expectations are over Brownian paths  $W_i$  only. For multi-valued initial conditions, expectations are over both initial conditions and Brownian paths.

A scheme is said to converge weakly with  $O(\Delta t_l^\alpha)$  if  $\varepsilon_W \leq c\Delta t_l^\alpha$ , and strongly with  $O(\Delta t_l^\beta)$  if  $\varepsilon_S \leq c\Delta t_l^\beta$  as  $\Delta t_l \rightarrow 0$ , where the  $c$ 's are (different) constants. While strong convergence is a straightforward generalization of deterministic numerical convergence, it is rarely of practical importance. In general, it is the weak convergence properties of a numerical SDE scheme that dictates its utility.

For the Euler-Maruyama scheme (7), the convergence properties are

$$P(\mathbf{v}) - P(\mathbf{v}_l) = \begin{cases} \varepsilon_W \sim O(T\Delta t_l) & - \text{Weak Euler scaling,} \\ \varepsilon_S \sim O(\sqrt{T\Delta t_l}) & - \text{Strong Euler scaling,} \end{cases} \quad (10)$$

so  $\alpha = 1$  and  $\beta = 1/2$ , as shown in Fig. 2.

The next scheme in the hierarchy of Taylor expansions of (5) is the first order Milstein approximation, also shown in Fig. 1. It is [37, 39]

$$\Delta v_i = F_i \Delta t_l + D_{ij} \Delta W_j + \frac{1}{2} D_{mj} \frac{\partial D_{ij}}{\partial v_m} (\Delta W_j^2 - \Delta t_l) + \sum_{j \neq k} D_{mk} \frac{\partial D_{ij}}{\partial v_m} A_{kj}, \quad (11)$$

where  $\Delta t_l$  arising in the third term comes from the quadratic variation of a stochastic random variable, and  $A_{kj}$  is the off-diagonal ‘area integral’ cross term given by

$$A_{kj} = \int_t^{t+\Delta t_l} [W_j(s) - W_j(t)] dW_k(s) = \int_t^{t+\Delta t_l} dW_k(s) \int_t^s dW_j(\eta). \quad (12)$$

The area integrals  $A_{kj}$  are non-Gaussian random numbers that are closely related to the so-called ‘Lévy areas’  $L_{kj} = (A_{jk} - A_{kj})/2$ , and are correlated with the Brownian motions  $W_k$  and  $W_j$  [40]. Numerically, sampling  $A_{kj}$  in a computationally efficient manner is technically challenging [37, 41, 42]. However, recently Dimits et al. [33] have developed a simple new approximate method for sampling  $A_{kj}$  in two dimensions. The method is simple to implement, inexpensive, accurate, and relies on the joint probability density function of the area integrals only. Using the prescriptions outlined in [42] and [43], it is expected that this two dimensional area integral can be used to generate the  $D(D-1)/2$  non-independent area integrals that arise in higher velocity-space dimensions  $D$ .

The weak and strong convergence properties of the first-order Milstein scheme (11) are

$$P(\mathbf{v}) - P(\mathbf{v}_l) = \begin{cases} \varepsilon_W \sim O(T\Delta t_l) & - \text{Weak Milstein scaling,} \\ \varepsilon_S \sim O(\sqrt{T\Delta t_l}) & - \text{Strong Milstein scaling,} \end{cases} \quad (13)$$

so  $\alpha = 1$  and  $\beta = 1$ , and, therefore, (11) is superior to (7) only in its strong convergence properties, Fig. 2.



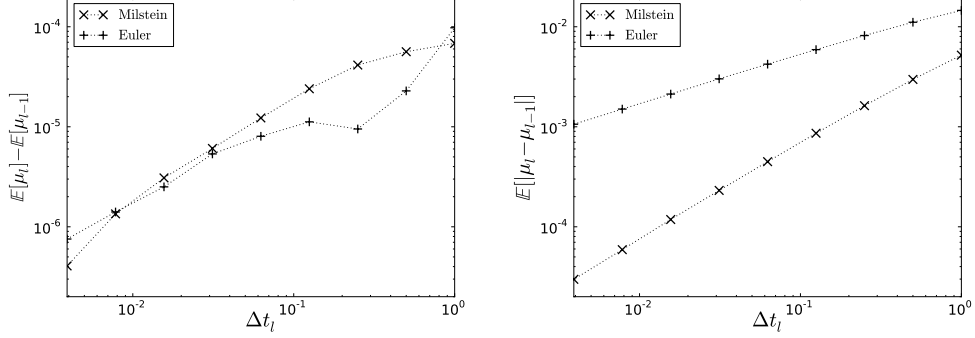


Figure 2: Weak (left) and strong (right) scaling properties of the Euler-Maruyama and Milstein schemes for the  $\mu$  component of the velocity calculated from  $10^8$  samples. While both schemes have the same order of weak convergence,  $O(\Delta t_l)$ , their strong convergence properties differ, (10) and (13). The Milstein scheme convergence strongly as  $O(\Delta t_l)$ , relative to the Euler-Maruyama scheme  $O(\sqrt{\Delta t_l})$ .

In the context of plasma physics, it is weak convergence that is typically important in simulating collisions *directly* using schemes like (7) and (11). This is because plasmas are many particle systems in which it is the summed distribution  $f$ , as opposed to the individual particles, that are important. In other words, particle identity, which is incorporated into the strong error, is unimportant in constructing and evolving the distribution function.

However, as we show in Section 3, it is the strong convergence properties of the underlying scheme that determines the computational efficiency of Giles' MLMC scheme. This is an instance of strong convergence being relevant to plasma physics. The MLMC method is significantly more efficient than direct methods, and especially so when used in conjunction with an underlying scheme with higher-order strong convergence. Quantitatively, the relationship between error, efficiency and computational cost, can be understood as follows.

### 2.3. Efficiency and computational cost

The expectation of the solution to (5),  $\mathbb{E}[\mathbf{v}(T)]$ , has two sources of error in its numerical realization. A finite-timestep error that depends on  $\Delta t_l$ , and a finite-sampling error that depends on the number of samples  $N$ . The same is true of any function of the solution, for example, the average kinetic energy  $\mathcal{K}$  of a collection of particles:

$$\mathcal{K} = \frac{1}{2} \frac{m}{n} \int f(T, \mathbf{v}) |\mathbf{v}|^2 d^3 \mathbf{v} \equiv \frac{1}{2} m \mathbb{E}[|\mathbf{v}(T)|^2], \quad (14)$$

where the left and right hand interpretations of  $\mathbf{v}$  are as in (4), so  $f$  obeys (1) and  $\mathbf{v}(T)$  obeys (5).

Minimizing the error in the moments of  $f$ , for example (14), is a compromise between efficiency and accuracy. Let  $P = P(\mathbf{v}(T))$  be some Lipschitz scalar function of  $\mathbf{v}(T)$ , let  $P_l = P(\mathbf{v}_l(T))$  be its finite timestep approximation, and let  $P_l^k = P(\mathbf{v}_l^k(T))$  be the  $k$ -th sample of the finite timestep approximation. For numerical schemes that employ discretizations like (7) or (11)

directly, we define

$$\hat{P} = \mathbb{E}[P] \quad \text{with } N \rightarrow \infty, \Delta t_l \rightarrow 0, \quad (15)$$

$$\hat{P}_l = \mathbb{E}[P_l] \quad \text{with } N \rightarrow \infty, \Delta t_l = 2^{-l}T, \quad (16)$$

$$\hat{P}_l^{N_l} = N_l^{-1} \sum_{k=1}^{N_l} P_l^k \quad \text{with } N = N_l, \Delta t_l = 2^{-l}T. \quad (17)$$

to be the ‘true’, finite-timestep, and finite-timestep finite-sampling approximations respectively, Table 1.

Equations (15)-(17) are calculated from (5) in two stages. First, applying some convergent integration scheme with  $\Delta t_l \rightarrow 0$  for  $\mathbf{v}$ , or  $\Delta t_l = 2^{-l}T$  a constant for  $\mathbf{v}_l$ . Second, applying  $P$  and calculating the expectation by generating multiple samples, and then averaging over them with  $N \rightarrow \infty$  for  $\hat{P}$  or  $\hat{P}_l$ , and finite  $N = N_l$  for  $\hat{P}_l^{N_l}$ .

An accurate estimate  $\hat{P}_l^{N_l}$  of  $\hat{P}$  is then one for which the mean squared error (MSE)

$$\begin{aligned} \text{MSE} &\equiv \mathbb{E} \left[ \left( \hat{P} - \hat{P}_l^{N_l} \right)^2 \right] \\ &= \left( \hat{P} - \hat{P}_l \right)^2 + \mathbb{E} \left[ \left( \hat{P}_l - \hat{P}_l^{N_l} \right)^2 \right], \end{aligned} \quad (18)$$

is small. The final equality follows from the fact that  $\mathbb{E}[\hat{P}_l - \hat{P}_l^{N_l}] = \hat{P}_l - \hat{P}_l^{N_l}$  and  $\hat{P}_l^{N_l}$  is an unbiased estimate of  $\hat{P}_l$ .

The size of the two terms in (18) can be varied independently. The first depends on the weak convergence rate of the scheme  $\mathcal{O}(\Delta t_l^\alpha)$ , and is independent of  $N$ . The second depends on the number of samples  $N = N_l$ , and its size is independent of  $\Delta t_l$ . Their associated sizes are

$$\left( \hat{P} - \hat{P}_l \right)^2 \lesssim c_1^2 \Delta t_l^{2\alpha}, \quad \mathbb{E} \left[ \left( \hat{P}_l - \hat{P}_l^{N_l} \right)^2 \right] = \frac{\text{Var}[P_l^k]}{N_l}, \quad (19)$$

where

$$\text{Var}[P] \equiv \mathbb{E}[(P - \mathbb{E}[P])^2]$$

is the variance operator on a random variable, and  $c_1$  is a constant.

It follows that  $\hat{P}_l^{N_l}$  is accurate to within  $\varepsilon$  of  $\hat{P}$  if

$$\varepsilon^2 \geq \text{MSE} \sim c_1^2 \Delta t_l^{2\alpha} + \frac{\text{Var}[P_l^k]}{N_l}. \quad (20)$$

The challenge in enforcing this bound is to do so as efficiently as possible.

For *direct* integration, the computational cost  $K$  of obtaining  $\hat{P}_l^{N_l}$  is the product of the number of timesteps  $T/\Delta t_l = 2^l$  and the number of samples  $N = N_l$ . It provides a simple measure of efficiency and is defined as

$$K = N_l \frac{T}{\Delta t_l}, \quad (21)$$

In practice,  $K$  measures the number of times the collision integration routine must be called in a numerical simulation.

Commonly used notation	
$\mathbf{v}, v_i$	Velocity vector, components
$\mathbf{v}^k, v_i^k$	$k$ -th realization of $\mathbf{v}, v_i$
$\mathbf{v}_l$	$\mathbf{v}$ calculated with $\Delta t_l$
$v, \mu, \phi$	Spherical components of $\mathbf{v}$
$\hat{\mathbf{v}}, \hat{v}_l$	$\mathbb{E}[\mathbf{v}], \mathbb{E}[\mathbf{v}_l]$
$P, P_l$	$P(\mathbf{v}), P(\mathbf{v}_l)$
$\hat{P}, \hat{P}_l$	$\mathbb{E}[P(\mathbf{v})], \mathbb{E}[P(\mathbf{v}_l)]$
$V_l$	$\text{Var}[P_l - P_{l-1}]$

Table 1: Roman sub- and superscripts, with the exception of  $l, N_l, L, N_L$ , are vector components and random variable realizations respectively.

To make the scheme as efficient as possible, we wish to minimize  $K$  subject to (20). To do so, we will ensure both terms in (18) are individually bounded, so their sum  $\leq \varepsilon^2$ . Applying the method of Lagrange multipliers in case of equality in (20) yields expressions for the optimal  $\Delta t_l$  and  $N_l$ :

$$\begin{aligned}\Delta t_l &\simeq \varepsilon^{1/\alpha} [c_1^2 (2\alpha + 1)]^{-1/2\alpha}, \\ N_l &\simeq \varepsilon^{-2} \left(1 + \frac{1}{2\alpha}\right) \text{Var}[P_l^k],\end{aligned}$$

Direct substitution into (21) then reveals the optimal computational cost is

$$K_{\text{opt}} \simeq \frac{c_2}{\varepsilon^{(2+1/\alpha)}} \frac{1}{2\alpha} \left(1 + \frac{1}{2\alpha}\right)^{1+1/2\alpha} \text{Var}[P_l^k], \quad (22)$$

where  $c_2$  is a constant.

It is important to note that for both the Euler-Maruyama and the first order Milstein schemes, (7) and (11),  $\alpha = 1$ . It follows that  $K_{\text{opt}} \sim \mathcal{O}(\varepsilon^{-3})$ . It is only by including higher order terms that the weak error scaling, and therefore the optimal computational cost, can be improved.

While the direct approach has the advantage of being conceptually simple, it is asymptotically inefficient. Minimizing the error using *direct* methods requires both a large sample size and a small step size, which tends to over-resolve the problem. It is this inefficiency that is improved upon by the MLMC method.

### 3. Multilevel Monte Carlo method

#### 3.1. Background

The computational cost of direct methods scales with their timestep resolution and expectation sample size. The improved efficiency of the MLMC method, relative to the methods in Section 2.2, comes from judiciously expending computational resources only when necessary. As initially described by Giles [7], and reviewed in this section, the improved efficiency of the method is achieved by building an estimate of  $\hat{P}$  from multiple solutions with varying timesteps  $\Delta t_l = 2^{-l}T$ , i.e. values of  $l$ , and expectations with varying sample sizes  $N_l$ . For the coarsest level  $l = 0$ , the Langevin equation is integrated with a single timestep, while for the finest level  $l = L$ ,  $2^L$  timesteps are required.

The basic mechanism behind the method's improved efficiency can be understood as follows. For small values of  $l$ , estimates are inexpensive to compute accurately, because only a few timesteps are required for each realization of the numerical solution. In turn, for large values of  $l$ , where each integration is relatively expensive, only a few realizations are needed because the finite-sampling error converges to zero as the strong error, assuming  $\beta$  is positive.

From the linearity of the expectation operator, we have the following identity

$$\begin{aligned}\hat{P}_L &\equiv \mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^L \mathbb{E}[P_l - P_{l-1}], \\ &\equiv \hat{P}_0 + \sum_{l=1}^L \delta \hat{P}_l\end{aligned}\tag{23}$$

where  $\hat{P}_0 = \mathbb{E}[P(v_0)]$  is estimated using a single timestep and  $\delta \hat{P}_l \equiv \mathbb{E}[P_l - P_{l-1}]$ . Equation (23) describes a telescoping sum, where the contribution of each term decreases with increasing  $l$ , as shown in Fig. 3.

The finite sampling analogue of (23) can be obtained by generating  $N_0, N_l$  samples of  $\hat{P}_0, \delta \hat{P}_l$  and combining them so

$$\hat{P}_L^{N_L} = \hat{P}_0^{N_0} + \sum_{l=1}^L \delta \hat{P}_l^{N_l}\tag{24}$$

where

$$\hat{P}_0^{N_0} = \frac{1}{N_0} \sum_{k=1}^{N_0} P_0^k,\tag{25}$$

$$\delta \hat{P}_l^{N_l} = \frac{1}{N_l} \sum_{k=1}^{N_l} (P_l^k - P_{l-1}^k),\tag{26}$$

are unbiased estimates of  $\hat{P}_0, \delta \hat{P}_l$  respectively.

In calculating each pair  $P_l^k$  and  $P_{l-1}^k$  that contributes to the sum in  $\delta \hat{P}_l^{N_l}$ , it is essential that the payoffs are constructed from the same underlying stochastic path and initial conditions. That is, for each contributing realization to  $\delta \hat{P}_l^{N_l}$ ,  $P_{l-1}^k$  must be constructed by suitably coarsening  $P_l^k$ , or conversely,  $P_l^k$  must be calculated by suitably refining  $P_{l-1}^k$ . One coarsening method, including a prescription for the multi-dimensional Lévy areas, is provided in section 4.3 of Dimits et al. [33]. Paths and initial conditions for different realizations that contribute to  $\delta \hat{P}_l^{N_l}$ , and indeed different  $\delta \hat{P}_l^{N_l}$ 's and  $\hat{P}_0^{N_0}$ , can be calculated independently.

Equation (24) returns a good estimate of  $\hat{P}, \hat{P}_L^{N_L}$ , if, for a reasonable computational cost, the total error is small. Like the direct methods of Section 2.3, the finite-timestep contribution to the total error is governed by the weak convergence properties of the underlying scheme. However, unlike direct methods and crucially for the MLMC method, the finite-sampling, or variance, contribution

$$\text{Var}[\hat{P}_L^{N_L}] = \text{Var}[\hat{P}_0^{N_0}] + \sum_{l=1}^L \text{Var}[\delta \hat{P}_l^{N_l}],$$

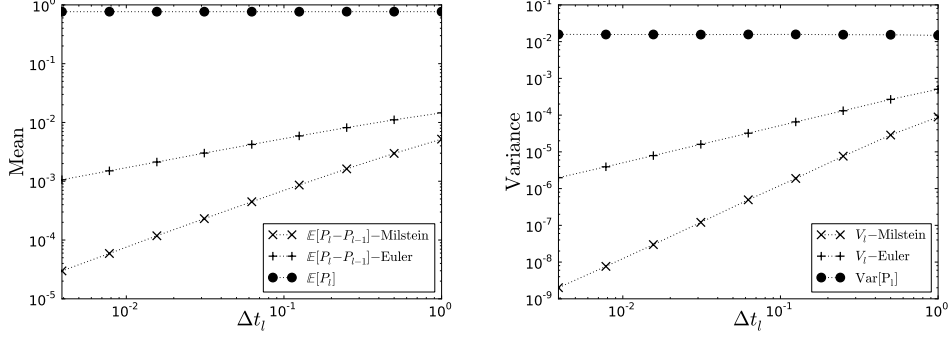


Figure 3: Mean (left) and variance (right) of the difference between levels for the Euler and Milstein schemes. The mean of the difference at level  $l$  appears explicitly in (24). The variance in differences at  $l$  is significantly less than that of a single level, which allows term in  $\hat{P}_L^{N_L}$  to be calculated efficiently using MLMC methods. Data is taken from the beam diffusion test case in Section 4 with payoff  $P = \mu$ , (delta-function) initial conditions  $v^* = 0.5, \mu^* = 0.8$ , and final time  $T = 0.02$ . All quantities calculated in this figure are taken using  $10^5$  samples.

is determined by the strong convergence properties of the underlying scheme.

As in (18), the mean square error is given by

$$\text{MSE} = (\hat{P} - \hat{P}_L)^2 + \mathbb{E} \left[ (\hat{P}_L - \hat{P}_L^{N_L})^2 \right], \quad (27)$$

which we wish to bound so that  $\varepsilon^2 \geq \text{MSE}$ . Analogous to (19), the two terms are of size

$$(\hat{P} - \hat{P}_L)^2 \lesssim c_1^2 \Delta t_l^{2\alpha}, \quad \mathbb{E} \left[ (\hat{P}_L - \hat{P}_L^{N_L})^2 \right] = \frac{V_0}{N_0} + \sum_{l=1}^L \frac{V_l}{N_l}, \quad (28)$$

where  $V_l \equiv \text{Var}[P_l^k - P_{l-1}^k]$  and  $V_0 = \text{Var}[V_0^k]$  are the variances of a single sample. The variances of these samples are related to those of the random variable  $\delta \hat{P}_l^{N_l}$  by  $\text{Var}[\delta \hat{P}_l^{N_l}] \simeq V_l/N_l$  and  $\text{Var}[\hat{P}_0^{N_0}] \simeq V_0/N_0$ . For  $l > 0$ ,  $V_l$  follows the strong convergence order of the underlying scheme (9):

$$V_l \lesssim c_3 \Delta t_l^{2\beta}. \quad (29)$$

where  $c_3$  is a constant. We demonstrate this result numerically in Fig. 3, and note that (29) dictates that the finite-sampling error in  $\delta \hat{P}_l^{N_l}$  can be bounded using fewer and fewer samples  $N_l$ , as  $l$  increases ( $\Delta t_l$  decreases).

From (27) and (28), it follows that  $\hat{P}_L^{N_L}$  is a good estimate of  $\hat{P}$  if

$$\varepsilon^2 \geq \text{MSE} \sim c_1^2 \Delta t_l^{2\alpha} + \frac{V_0}{N_0} + \sum_{l=1}^L \frac{V_l}{N_l}, \quad (30)$$

which has an associated computational cost of

$$K = \sum_{l=0}^L K_l \equiv \sum_{l=0}^L N_l \frac{T}{\Delta t_l}. \quad (31)$$

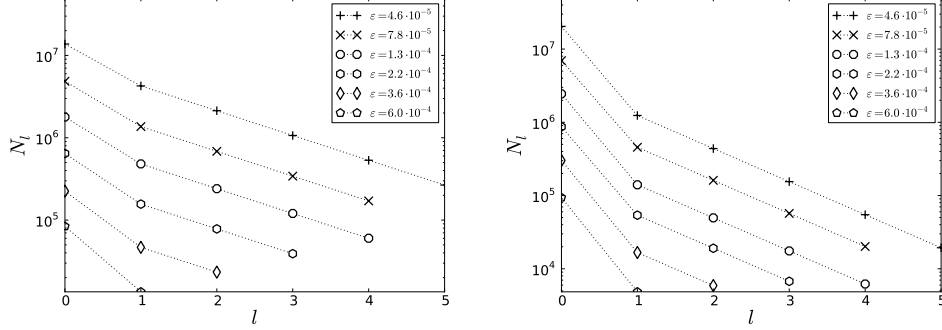


Figure 4: Samples  $N_l$  at each level  $l$  for the MLMC scheme with Euler (left) and Milstein (right) discretizations, (34). The scaling for levels  $l \geq 1$  is determined by the strong convergence properties of the underlying scheme,  $N_l \sim 2^{-l\beta}$ . The computational cost at each level  $K_l = N_l 2^l$  is approximately constant for the Euler method, but decreases rapidly in the telescoping Milstein sum. Parameters are the same as those used in Fig. 3.

The most efficient method for calculating  $\hat{P}_L^{N_L}$  is, again, the one that minimizes  $K$  subject to (30). Unlike *direct* methods, there are now two new degrees of freedom over which to optimize: the total number of levels  $L$ , and the number of samples used for the expectation at each level  $N_l$ . As in the previous section, the minimal  $K$  will clearly occur when  $\text{MSE} = \varepsilon^2$  and so we approach the problem by separately bounding the two terms in (27) as follows:

$$(\hat{P} - \hat{P}_L)^2 = \frac{1}{2}\varepsilon^2, \quad \mathbb{E}[(\hat{P}_L - \hat{P}_L^{N_L})^2] = \frac{1}{2}\varepsilon^2. \quad (32)$$

The first condition, along with (30) gives

$$L = \frac{1}{\alpha} \frac{\ln[c_1 T^\alpha \sqrt{2}\varepsilon^{-1}]}{\ln 2}. \quad (33)$$

Considering this  $L$  fixed, a Lagrange multiplier argument reveals the optimal efficiency is obtained when  $N_l \sim \sqrt{V_l T 2^{-l}}$ . Using this and the second condition in (32), the optimal number of samples at level  $l$  is given by

$$N_l = \frac{\sqrt{V_l 2^{-(l+2)}}}{\varepsilon^2} \sum_{l=0}^L \sqrt{V_l 2^l}, \quad (34)$$

where (29) ensures that  $N_l$  is a strictly decreasing function of  $l$  as shown in Fig. 4.

Now, combining (29), (31), (33) and (34), the optimal computational cost of the MLMC scheme is given by

$$K_{\text{opt}} \simeq \frac{2c_4 T^{(2\beta-1)/2}}{\varepsilon^2} \left( \sum_{l=0}^L 2^{-l(2\beta-1)/2} \right)^2, \quad (35)$$

where  $L = L(\varepsilon)$  is given by (33). In the case of  $\beta = 1/2$ , the sum in (35) scales as  $L \sim \ln \varepsilon$ , whereas for  $\beta > 1/2$ , the sum can be uniformly bounded. From this, the asymptotic cost of

the MLMC method is  $O(\varepsilon^{-2}(\ln \varepsilon)^2)$  for the Euler-Maruyama method and  $O(\varepsilon^{-2})$  for the Milstein method.

These costs are to be contrasted with the total cost of direct and binary methods, for which  $K$  is given by (21). As described in Section 2.3, the computational cost of these methods can be easily calculated by writing the requisite (so that the  $\text{MSE} \leq \varepsilon^2$ ) timestep  $\Delta t_l$  and sample size  $N$  in terms of  $\varepsilon$  and substituting directly into (21). For direct methods, the result of doing so is given by (22) so  $K \sim O(\varepsilon^{-(2+1/\alpha)})$  for a general weak order- $\alpha$  scheme, and  $K \sim O(\varepsilon^{-3})$  for the widely used  $\alpha = 1$  direct Euler-Maruyama integration scheme. For binary methods, the analysis is identical<sup>2</sup>. The finite-timestep error is, at best,  $O(\Delta t_l)$  and the finite-sampling error is  $O(N^{-1/2})$ , so the requisite scalings of these two terms are  $\Delta t_l \sim O(\varepsilon)$  and  $N \sim O(\varepsilon^{-2})$  respectively. It follows that for the binary method, at best,  $K \sim O(\varepsilon^{-3})$  and, at worst, when the finite-timestep error is  $O(\Delta t_l^{1/2})$ , the cost is  $K \sim O(\varepsilon^{-4})$ .

The relative theoretically optimal costs of the various methods are therefore:

$$K_{\text{opt}} = \begin{cases} O(\varepsilon^{-3}) & - \text{ Binary collisions,} \\ O(\varepsilon^{-(2+1/\alpha)}) & - \text{ General order-}\alpha \text{ direct SDE,} \\ O(\varepsilon^{-2}(\ln \varepsilon)^2) & - \text{ MLMC with } \beta = 1/2, \\ O(\varepsilon^{-2}) & - \text{ MLMC with } \beta > 1/2. \end{cases} \quad (36)$$

In Fig. 5 we consider the specific test case of the collisional relaxation of a monoenergetic, low-density beam, as described in Section 4. The figure confirms that the cost scaling in (36) are accurate for the  $\alpha = 1$  direct Euler method, and the Euler and Milstein multilevel schemes. It also shows that the computational cost of the MLMC method is substantially less than that of the direct method.

### 3.2. Numerical Implementation

Equations (24)-(26), (33) and (34) provide a prescription for approximating  $\hat{P}$  by  $\hat{P}_L^{N_L}$ , such that its  $\text{MSE} \leq \varepsilon^2$ . There are two degrees of freedom in the MLMC scheme,  $L$  and  $N_L$ , each influencing an associated finite-timestep and finite-sampling error. The constants that determine  $L$  and  $N_L$ , such that (32) is enforced, are  $c_1$  and  $V_L$  respectively.

In the asymptotic limit of small timestep (large  $l$ ),  $c_1$  is the constant of proportionality between  $\hat{P} - \hat{P}_l$  and  $\Delta t_l$ , as defined in the *weak* error (8). It can be calculated using Richardson extrapolation

$$|c_1| \simeq |c_1|_{(N)} \equiv \frac{|\hat{P}_l^N - \hat{P}_{l-1}^N|}{T^\alpha 2^{-l\alpha} |1 - 2^\alpha|}, \quad (37)$$

where  $P_l^N, P_{l-1}^N$  are determined empirically by direct integration with the relevant discretization (Euler-Maruyama, first-order Milstein) and the integer  $N \gg 1$  is large enough that the sampling

<sup>2</sup>Note that binary collision algorithms pair particles into  $N/2$  sets when performing collisions. This offers a relative saving of up to a factor of a half, compared to Langevin treatments [44], although the constant factor does not affect the scaling properties of the algorithm.

error in  $\hat{P}_l^N$  is small relative to the timestep error i.e.  $\hat{P}_l^N \simeq \hat{P}_l$ . This semi-equality can be checked, *ex post facto*, by ensuring that

$$1 \gg \frac{|c_1|_{(nN)} - |c_1|_{(N)}}{|c_1|_{(nN)}},$$

for  $n > 1$ , also an integer.

As for  $c_1$ ,  $V_L$ , which influences the finite sampling error in the MLMC scheme, must be determined empirically. It can be estimated by taking  $N$  samples of  $P_L - P_{L-1}$ , and setting

$$V_L \simeq V_L^N \equiv \frac{1}{N} \sum_{k=1}^N (P_L^k - P_{L-1}^k)^2 - \frac{1}{N^2} \left( \sum_{k=1}^N (P_L^k - P_{L-1}^k) \right)^2, \quad (38)$$

where each value of  $P$  in the pair is generated from the same Brownian path and initial condition using the relevant discretization, and  $N$  should be large enough to ensure good statistics. It is important to note that, unlike (37), this quantity depends on the *strong* convergence properties of the underlying integration scheme (9). In this case,  $P_L$  and  $P_{L-1}$  must be calculated using different timesteps, but the same underlying stochastic path in which the path at the coarser level  $L-1$  is suitably compounded from those used at the finer level  $L$ . Using  $V_L$ , the number of samples at each level  $N_l$  can then be computed according to (34) and by noting  $V_l = V_L 2^{2\beta(L-l)}$ .

The  $l = 0$  level is an exception and, analogous to (38), it is given by

$$V_0 \simeq V_0^N \equiv \frac{1}{N} \sum_{j=1}^N (P_0^j)^2 - \frac{1}{N^2} \left( \sum_{j=1}^N P_0^j \right)^2, \quad (39)$$

where, again,  $N$  should be sufficiently large.

Careful calculation of the constants in this section is essential to obtaining an accurate estimate of  $\hat{P}_L^{N_L}$  using the MLMC method<sup>3</sup>. (Although it should be noted that even for direct methods,  $c_1$  must still be calculated to ensure  $\text{MSE} \leq \varepsilon^2$ .) The method can be implemented, step by step, as follows, where special note should be taken at step 6 where it is essential that each pair of realization be calculated consistently from the same underlying path. A prescription for doing so is provided in [33].

The steps are:

1. Choose a payoff function  $P$ , end time  $T$ , and an acceptable error bound  $\varepsilon$ .
2. Choose a method of direct integration for the MLMC method.
3. Use (37) to calculate  $c_1$ , and combine with (33) to get  $L$ .
4. Use  $L$ , (38) and (39) to calculate  $V_L$  and  $V_0$ .
5. Use  $V_L$ ,  $V_0$  and (34) to calculate  $N_l$  and  $N_0$ .
6. Calculate  $N_l$  pairs of  $P_l^k, P_{l-1}^k$ , each with the same underlying stochastic path.
7. Use  $N_l$  pairs of  $P_l^k, P_{l-1}^k$  to calculate  $\delta \hat{P}_l^{N_l}$  for each  $l = 1$  to  $L$ .
8. Use  $N_0$  to calculate  $P_0^{N_0}$ .
9. Use  $P_0^{N_0}$  and  $\delta \hat{P}_l^{N_l}$  from  $l = 1$  to  $L$  to calculate  $\hat{P}_L^{N_L}$  according to (24).

These steps are implemented in Section 4 for a test case describing the collisional diffusion of a beam of a particles interacting with a Maxwellian background.

<sup>3</sup>An alternative approach to bounding the bias error is based on increasing  $L$  until the condition  $|\delta \hat{P}_L^{N_L}| < \varepsilon / \sqrt{2}$  is met [7]. In the case of a sign change between successive  $\delta \hat{P}_L^{N_L}$ , modifications are required.



#### 4. Beam Diffusion Test Case

The average pitch-angle evolution of a spatially homogeneous, gyrotropic beam of particles  $a$  constitutes a simple and robust test case for the MLMC method. The beam is injected into a Maxwellian background distribution of particles  $b$  with equal mass,  $m_a = m_b$ . In the absence of forcing, the action of collisions isotropizes  $f_a$  in this classical relaxation problem.

Working in spherical velocity-space coordinates  $(v, \theta, \phi)$ ,  $v$  is the particle speed,  $\theta = \cos^{-1} \mu$  is the pitch angle with respect to some preferred direction  $\mu$ , and  $\phi$  is the azimuthal angle. Neglecting  $\phi$ -dependence (i.e. a two-dimensional collision model), the collision operator (1) is [35]:

$$\frac{\partial f_a}{\partial t} \Big|_{\text{coll}} = -\frac{1}{v^2} \frac{\partial}{\partial v} \left[ v^2 \frac{\partial h}{\partial v} + \frac{\partial g}{\partial v} f_a \right] + \frac{1}{2v^2} \frac{\partial^2}{\partial v^2} \left( v^2 \frac{\partial^2 g}{\partial v^2} f_a \right) + \frac{1}{2v^2} \frac{\partial g}{\partial v} \frac{\partial}{\partial \mu} \left[ (1 - \mu^2) \frac{\partial f_a}{\partial \mu} \right], \quad (40)$$

and the (initial) particle distributions are

$$f_a = n_a \delta(\mathbf{v}^* - \mathbf{v}), \quad (41)$$

$$f_b = \frac{n_b}{(\pi v_{\text{th},b}^2)^{3/2}} \exp[-v^2/v_{\text{th},b}^2], \quad (42)$$

where  $\mathbf{v}^*$  is some single valued initial velocity for the test particles and the Maxwellian field particle thermal velocity is  $v_{\text{th},b}^2 = 2\tau/m_b = (2/3n_b) \int f_b |\mathbf{w}|^2 d^3 \mathbf{v}$  where  $\tau$  is the temperature of  $f_b$  and  $\mathbf{w} = \mathbf{v} - \mathbf{u}$  is the random, i.e. particle minus flow, velocity. We set  $n_b \gg n_a$  throughout so we can neglect the back reaction of the beam on the Maxwellian.

In the case that  $f_b$  is Maxwellian, Trubnikov [45] gives  $g, h$  concisely by:

$$g(v) = \frac{1}{2} \Gamma n_b \sqrt{2} v_{\text{th},b} \left[ \Phi \left( 2x + \frac{1}{x} \right) + \Phi' \right], \quad (43)$$

$$h(v) = 2\Gamma n_b \frac{\Phi}{v}, \quad (44)$$

where  $x = v/v_{\text{th},b}$ , and  $\Phi(x)$  is the standard error function.

The set of Langevin equations (5) corresponding to (40)-(44) are then given by

$$dv(t) = F(v)dt + \sqrt{D_v(v)} dW_v(t), \quad (45)$$

$$d\mu(t) = -2D_a(v)\mu dt + \sqrt{2D_a(v)(1 - \mu^2)} dW_\mu(t), \quad (46)$$

where  $D_v, D_a$  are the speed and angular diffusion coefficients, and we have normalized  $f_a$  by  $2\pi v^2$  as required to bring the derivative to the outside in (40), so in (45) and (46),  $v(t) \rightarrow 2\pi v^3(t)$  and  $\mu(t) \rightarrow 2\pi\mu(t)v^2(t)$  in un-normalized coordinates. In what follows we work in normalized coordinates.

The curvilinear coordinate system requires the coefficients of the deterministic and stochastic terms to be a mixture of  $F, D_a, D_v$ , and these are given by [33]

$$F(v) = -\frac{A_D}{2v^2} [(4x + 1)G(x) - \Phi(x)], \quad (47)$$

$$D_v(v) = \frac{A_D}{2v} G(x), \quad (48)$$

$$D_a(v) = \frac{A_D}{4v^3} (\Phi(x) - G(x)), \quad (49)$$

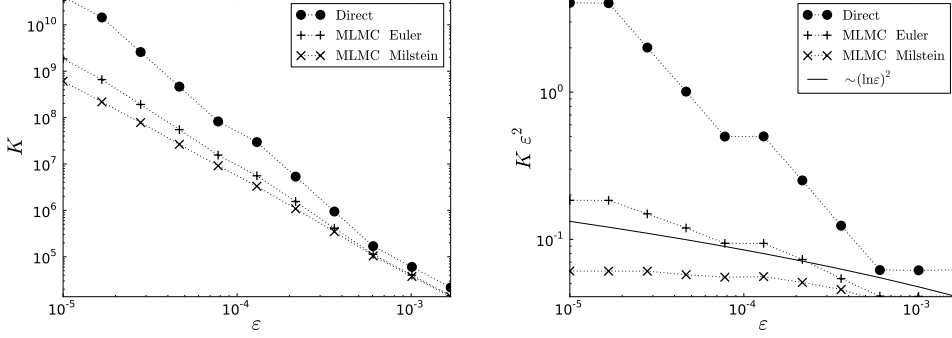


Figure 5: Computational cost  $K$  (left) and normalized  $K$  (right) versus user-prescribed error bound  $\varepsilon$  for the beam diffusion test case in Section 4. The parameters are as in Fig. 3. Both the Euler and Milstein MLMC schemes are more efficient than direct integration, Milstein by a factor of approximately 100 for the case  $\varepsilon = 10^{-5}$ , and the scaling costs predicted by (36) are recovered.

where  $A_D = 2n_b\Gamma = 8\pi n_b q_a^2 q_b^2 \Lambda / m_a^2$ , and  $G(x) = (\Phi - x\Phi')/2x^2$  is the Chandrasekhar function.

The finite-timestep discretized Langevin equations can then be obtained from (45) and (46) by an iterative stochastic Taylor expansion in  $\Delta t_l$ . To lowest order, retaining terms up to order  $O(\Delta t_l^{1/2})$ , we obtain the Euler-Maruyama scheme, and to next order, retaining terms up to  $O(\Delta t_l)$ , we obtain the Milstein scheme. Normalizing time  $t$  by the thermal field-particle collision rate  $\nu_b = \sqrt{2}A_D/\nu_{th,b}^3$  and velocity  $v$  by  $\sqrt{2}\nu_{th,b}$ , the dimensionless discretized Langevin equations are

$$\Delta v = F\Delta t_l + \sqrt{2D_v}\Delta W_v + \kappa_M D'_v \frac{1}{2} (\Delta W_v^2 - \Delta t_l), \quad (50)$$

$$\begin{aligned} \Delta \mu = & -2D_a\mu\Delta t_l + \sqrt{2D_a(1-\mu^2)}\Delta W_\mu + \\ & \kappa_M \left[ -2D_a\mu \frac{1}{2} (\Delta W_\mu^2 - \Delta t_l) + \sqrt{\frac{D_v}{D_a}} \sqrt{1-\mu^2} D'_a A_{v\mu} \right], \end{aligned} \quad (51)$$

where  $\kappa_M = 0, 1$  for Euler and Milstein respectively and  $A_{v\mu}$  is the  $v$ - $\mu$  correlated random variable (12) whose approximate characteristic function (the Fourier transform of its probability density function) is given in [33]. The coefficient functions are to be evaluated at the start of each timestep, as required by the Ito stochastic calculus, and the normalized drag and diffusion coefficients become

$$F(v) = 2v(D_a(v) - D_v(v)), \quad (52)$$

$$D_v(v) = \frac{1}{v} G\left(\frac{v}{\sqrt{2}}\right), \quad (53)$$

$$D_a(v) = \frac{1}{2v^3} \left[ \Phi\left(\frac{v}{\sqrt{2}}\right) - G\left(\frac{v}{\sqrt{2}}\right) \right]. \quad (54)$$

In equilibrium, where  $f_a$  and  $f_b$  have the same flow velocity  $\int f_a v d^3v/n_a = \int f_b v d^3v/n_b$  and temperature,  $(m_a/3n_a) \int f_a \text{Trace}(\mathbf{w}\mathbf{w}) d^3v = \tau$ , the drag and diffusion coefficients are related by

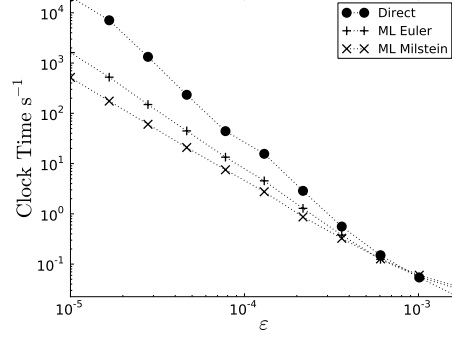


Figure 6: Wall clock time to execute steps 1-9 of Section 3.2 versus user prescribed error bound  $\varepsilon$  for the same parameters as in Fig. 3. The numerical code is written in Python and Fortran 90, and executed on a 2.4 GHz Intel Core i5 MacBook. The MLMC methods are significantly faster than the direct methods for high accuracy simulations. For small values of  $\varepsilon = 10^{-5}$ , the Milstein method is approximately 40 times faster than the direct alternative.

the spherical coordinate form of the Einstein relations (6). That is, (52)-(54) must obey

$$2D_a(v)\mu - \frac{\partial}{\partial\mu} [D_a(v)(1 - \mu^2)] = 0, \quad (55)$$

$$F(v) - \frac{1}{v} \frac{\partial}{\partial v} (v^2 D_v(v)) + v D_v(v) = 0, \quad (56)$$

which can be readily confirmed by direct substitution<sup>4</sup>.

To ensure the coefficients of the discretized Langevin equation are Lipschitz continuous, as required for the MLMC method, (50)-(54) must be numerically regularized upon implementation. The procedure for doing so is described in Appendix Appendix A.

Equations (50)-(54) constitute the building blocks for a MLMC scheme that returns the mean or moment of some payoff  $P$  of  $\mathbf{v}$  associated with  $f_a$  as it interacts with  $f_b$ . The building blocks are independent of  $P$ , the time at which its mean is evaluated  $T$ , and the acceptable error bound  $\varepsilon$ . These quantities are parameters of the simulation. Collectively they determine the preconditioning parameters of the method,  $c_1$ ,  $V_L$  and  $V_0$ .

We choose our payoff function

$$P = \mu,$$

so that the MLMC scheme approximates its mean value over all particles

$$\hat{P}(T) \simeq \hat{P}_L^{N_L}(T) \simeq \frac{1}{n_a} \int \mu f_a(T) d^3 \mathbf{v}. \quad (57)$$

The results of numerical implementation are shown in Figs. 5, 6 and 7 where the method successfully approximates the right hand side of (57), the ‘true’ value of which is itself approximated using a high-resolution many particle direct simulation Monte-Carlo scheme (the direct

<sup>4</sup>In addition to the Einstein relations,  $D_a$  and  $D_v$  satisfy  $D_v = d/dv(v^3 D_a)$  which, fundamentally, is a consequence of the fact that Coulomb interactions are through a central force and the Landau-Fokker-Planck treatment keeps only small-angle scattering interactions.

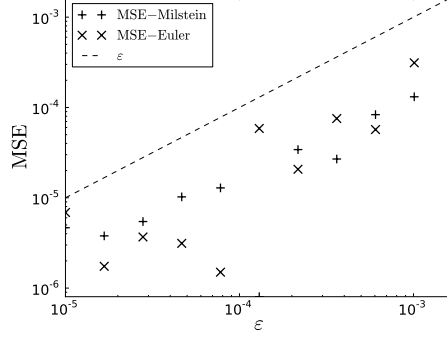


Figure 7: Mean Squared Error (MSE) vs user prescribed error bound  $\varepsilon$  for a range of MLMC runs with both Euler and Milstein (27). The MLMC method accurately approximates the true payoff to within  $\varepsilon$ , as all points fall below the delimiting dashed line. Parameters are as in Fig 3 and the mean is taken over 10 independent MLMC runs.

Euler scheme with  $2 \cdot 10^9$  particles and  $2^8$  timesteps.). For the most accurate case  $\varepsilon = 10^{-5}$ , both MLMC methods are considerably faster than the direct method for which the parameters (timestep, sample size) are chosen such that the  $\text{MSE} \leq \varepsilon^2$ . The Milstein MLMC method is approximately 100 times faster than the direct method in terms of its computational cost, Fig. 5, and 40 times faster in terms of its wall clock timing, i.e. the number of seconds required to complete the computation on a computer, Fig. 6.

The difference between the computational cost and wall clock timing arises, in part, because the MLMC method actually performs two integrations at each level  $l$ , a coarse and a fine integration. This leads to an additional cost of  $3/2$  not captured by (31). Furthermore, the MLMC Milstein method contains additional terms that include the Lévy areas. These must be calculated at an additional cost relative to the direct and Euler MLMC methods.

The MLMC scheme accurately (to within  $\varepsilon$ ) describes the average pitch angle relaxation of a beam of particles interacting with an isotropic Maxwellian background. The scaling predictions given by (36) are reproduced, so that the computational cost (total number of operations) of the Milstein, Euler and direct methods scale as  $\varepsilon^{-2}$ ,  $(\ln \varepsilon)^2 \varepsilon^{-2}$ , and  $\varepsilon^{-3}$  respectively. Integral to the Milstein MLMC method for  $P = \mu$  is an accurate description of  $A_{v\mu}$ , (51). For other payoffs in two dimensions that are independent of  $\mu$ , cross-terms like  $A_{v\mu}$  are not required. For example,

$$P = \frac{1}{2} m_a v^2$$

approximates the mean kinetic energy per particle  $\mathcal{K}$

$$\hat{P}(T) \simeq \hat{P}_L^{N_L}(T) \simeq \frac{1}{2} \frac{m_a}{n_a} \int f_a(T, \mathbf{v}) |\mathbf{v}|^2 d^3 \mathbf{v},$$

where the underlying Milstein scheme includes additional terms proportional to  $\Delta W_v^2$  and  $\Delta t_l$  only.

## 5. Discussion

The MLMC method constitutes a powerful new technique for solving kinetic expectation problems, the solutions to which can be used to reconstruct the underlying distribution function.

Asymptotically, it has significantly improved scaling properties compared to both direct SDE and binary collision methods. However, the method is limited in several respects.

Firstly, for any given problem, the multiplicative constant associated with the scaling of the computational cost may make the method prohibitively expensive. Secondly, it is unclear how, for  $t < T$ , mean field quantities like electromagnetic fields and evolving back-reacted drag and diffusion coefficients that dictate particle trajectories, are to be computed. Thirdly, for strongly non-equilibrium problems, a large number of moments or binning operations may be needed to accurately reconstruct  $f$ . This could be expensive.

In this section we discuss techniques and extensions to the MLMC method that can be used to address these problems.

### 5.1. Improved efficiency

For expediency, our description of the MLMC method presented in Section 3 was basic and concise. However, several improvements, not described earlier, exist.

Throughout, we have set the refinement factor between levels  $M = \Delta t_l / \Delta t_{l+1} = 2$ . Giles [7] has argued that while this choice is optimal for multilevel elliptic PDE solvers, for the MLMC method, other values may improve efficiency - specifically, a factor two saving may be obtained by setting  $M = 7$  in the MLMC Euler scheme. Less extreme values,  $M = 3, 4$  etc, also lead to improvements. Similarly, multiplicative constants other than a half for the finite timestep and finite sampling bounds,  $\varepsilon^2/2$ , in (32) may also lead to improvements.

Adaptive algorithms, and quasi-Monte Carlo sampling can also increase the efficiency of the method [46, 47, 48]. In particular, for the Milstein method, the rapid diminishing of the finite-sampling error with decreasing  $\Delta t_l$  means that the vast majority of the computational effort is expended at the coarsest  $l = 0$  level - Figure 4. Quasi-Monte Carlo methods are well suited to reducing this cost. Moreover, when the function  $P$  is sufficiently smooth (twice differentiable) - as would be expected in many plasma physics applications - the cost of simulating the coarsest level may be completely eliminated using the recently developed Ito linearization technique [49].

For existing code bases that implement the direct Euler scheme, it is possible to obtain the optimal  $\varepsilon^{-2}$  scaling of the MLMC scheme without simulating the Lévy areas. Using antithetic techniques, the sum in (35) can be bound, even though the underlying integrator has  $\beta = 1/2$  [39]. Generalizations of the antithetic method exist [49, 50].

Finally, because sampled paths are independent, like direct simulation Monte Carlo, the method can be readily parallelized. Indeed, the timescale over which the sampled paths of (5) can be integrated independently is only limited by the requirement that the mean fields are updated.

### 5.2. Mean fields

When the macroscopic dynamics of a system and its collisions occur on the same timescale, direct simulation Monte Carlo and particle-in-cell codes must perform two operations at each numerical timestep [29]. The first, is to advance the particles' positions  $\mathbf{x}$  and velocities  $\mathbf{v}$  according to the discretized equations of motion, including the mean force fields and collisions. In Langevin form, including spatial dependence and electromagnetic forces, the equation of motion (5) generalizes to [51]:

$$dx_i = v_i dt \tag{58}$$

$$dv_i = \left\{ \frac{e}{m} \left[ E_i + c^{-1} \epsilon_{ijk} v_j B_k \right] + F_i \right\} dt + D_{ij} dW_j, \tag{59}$$

where  $c$  is the speed of light,  $\epsilon_{ijk}$  is the Levi-Civita symbol, and  $E_i, B_i$  are the mean electric and magnetic fields. To ensure particle trajectories are calculated accurately over an extended period, a second operation must then be performed - the mean fields must be updated based on the new particle positions and velocities.

The mean electromagnetic fields,  $E_i, B_i$  are functions of  $t, \mathbf{x}$  only. They can be calculated from Maxwell's equations, a set of coupled first order linear PDEs, in terms of the sum over species of the macroscopic charge density  $\rho = en$  and current  $J_i = enu_i$  at each point in space.

In the Langevin framework, the macroscopic quantities can be accurately calculated on a timescale  $T$  using the MLMC scheme and an appropriate choice of payoff. For  $\rho$ , the payoff is

$$P = en\Theta(\mathbf{x}'), \quad (60)$$

where

$$\Theta = \begin{cases} 1, & \mathbf{x}' < \mathbf{x} < \mathbf{x}' + \delta\mathbf{x}, \\ 0, & \mathbf{x}' > \mathbf{x} > \mathbf{x}' + \delta\mathbf{x}, \end{cases} \quad (61)$$

is a binning function<sup>5</sup> for the real-space grid cell at position  $\mathbf{x}'$  and of size  $\delta\mathbf{x}$ , and  $n$  is the initial macroscopic density, as in (41). It follows that  $\rho(T, \mathbf{x}')$  is given by

$$\hat{P}(T) \simeq \hat{P}_L^{N_L}(T) \simeq e \int f(T, \mathbf{x}', \mathbf{v}') d^3\mathbf{v}' \quad (62)$$

and similarly for  $J_i$  with the payoff  $P = en\Theta(\mathbf{x}')v_i$ .

Along with Maxwell's equations and the equations for  $F, D$  in section 2.1, (58)-(62) provides a complete, efficient plasma description using MLMC methods. The description is, however, only efficient *and* accurate when it is acceptable to resolve the coefficients of (59), including  $F$  and  $D$ , self-consistently, on a slow timescale  $T$ . That is, if the inherent timescale on which the macroscopic mean fields evolve is  $\ll T$ , the MLMC method will fail to capture important dynamics that take place on this faster timescale. In its present form, the method is therefore restricted to the small Knudsen number regime where the collisional dynamics occur on a faster timescale than the macroscopic dynamics<sup>6</sup>. Within this framework, the MLMC method could itself constitute a building block for a multiscale simulation in which the collisions and macroscopic dynamics are resolved on timescales of  $O(\Delta t_l)$  and  $O(T)$  respectively.

It remains an open challenge to extend the MLMC method to kinetic problems in which there is no clear scale separation between the collisional and macroscopic dynamics i.e. Knudsen numbers of order one and greater. However, if this challenge could be met, it would constitute a potential game changer for kinetic plasma simulations in general.

### 5.3. Distribution functions

Thermalized distributions vary smoothly on a velocity-space scale  $v_{th}$  and can be uniquely determined from their first three moments. For non-equilibrium distributions, this is not the case. Two simple methods for reconstructing non-thermal  $f(T)$  using the MLMC method exist.

<sup>5</sup>Formally, the MLMC method requires  $P$  to be Lipschitz, and so simple step functions are inappropriate. Modifications to  $\Theta$  to ensure Lipschitz continuity may be necessary, but, nevertheless, improved efficiency relative to direct methods has been shown for a variety of non-Lipschitz payoffs [52].

<sup>6</sup>The method can also be applied when the Lorentz force is absent or externally imposed, and  $n_i \ll n_f$  or the collisions are between electrons and ion. In this case, the back reaction of the test particles can also be neglected.

The first is by summing the moment hierarchy, where each moment is calculated using the MLMC method with an appropriate choice of payoff  $P$ . For a complete set of moments, the structure of  $f$  can be captured identically (Chapter 7, [53]). For a finite-subset, as is practically achievable, an accurate approximation can still be obtained [54].

The second method for determining  $f$  is through a generalized version of (60) that bins particles in both real and velocity space<sup>7</sup>. In this case  $P = n\Theta(\mathbf{x}', \mathbf{v}')$  and  $\Theta$  is a simple generalization of (61) to include velocity space cells  $\mathbf{v}'$  of size  $\delta\mathbf{v}$ . It follows that  $\hat{P}(T) \simeq \hat{P}_L^{N_L}(T) \simeq f(T, \mathbf{x}', \mathbf{v}')$  returns the particle density in a phase space cell  $\mathbf{x}', \mathbf{v}'$  at time  $T$ .

Returning multiple outputs from a single run is useful for both the methods above [8]. Statistical errors withstanding, the same set of paths is needed to compute both successive moments and the binned phase-space distribution. So, by storing and re-using paths, the computational cost of calculating multiple moments is only approximately as much as the most expensive moment. The same is true for phase-space binning.

So far our discussion has focused on calculating distributions at time  $T$ . While extending the method to multi-valued initial conditions, unlike those in Section 4, is not technically difficult, a number of comments are in order.

First, chaotic particle trajectories, real (e.g. tokamak wall) and velocity space (e.g. magnetic mirror) boundaries are ubiquitous in plasma physics. Particles with nearby initial conditions, sampled from the same spatial cell, may drastically diverge in phase space. The consequences of this for the MLMC method are unknown.

Second, multi-valued initial conditions introduce a second source of statistical error, beyond that attributable to Brownian motion. When the variance in the initial data is much less than that associated with the random walk  $\text{Var}[v(0)] \ll \text{Var}[v(T)]$ , the computational cost of the simulation is unchanged. However, when the converse is true, the cost may increase dramatically. The ratio of the two terms is approximately

$$\frac{\text{Var}[v(T)]}{\text{Var}[v(0)]} \sim \frac{Te^4}{m_a^2} \frac{n_b v_{\text{th},b}^2}{n_a v_{\text{th},a}^2},$$

where we have assumed that the initial conditions for  $v(T)$  in the numerator are single-valued, i.e. given by (41), and that  $T \ll$  the macroscopic timescale.

## 6. Conclusion

For the first time, we have shown how the multilevel Monte Carlo integration scheme can be used to simulate Coulomb collisions in a plasma. Asymptotically, the method is up to  $\varepsilon^{-1}$  times faster than standard direct simulation Monte Carlo or binary collision methods, when used with an underlying Milstein discretization. This is illustrated in Fig. 5 where the total computational cost (operations count) for the direct SDE and the Euler and Milstein multilevel schemes are shown to scale as predicted. We have also demonstrated that the multilevel schemes are significantly faster than direct SDE methods in terms of both computational cost, Fig. 5, and wall clock time, Fig. 6. Our numerical results are for a classic beam diffusion test case in  $2D$  and over a given range of prescribed errors

---

<sup>7</sup>This method is qualitatively similar to an inverse semi-Lagrangian process [55], and comes at an informational cost equivalent to that incurred in a resampling procedure.

The most important extension to this work would be an expansion of the method to arbitrary Knudsen number problems, where a separation of collisional and macroscopic timescales does not exist. Other valuable studies would also include a demonstration of the method in forced, spatially inhomogenous and multi-physics problems, and an extension to kinetic collisions models other than the Coulomb case, for example neutral particle collisions.

#### *Acknowledgements*

Special thanks to the LLNL Visiting Scientist Program for hosting M.S.R. and L.F.R., and to B. Albright, S. Brunner, A. Cerfon, L. Chacón, F. Fiúza, M. Giles, M. Landreman, T. Wood, and B. Yan for helpful discussion throughout. This work was performed by UCLA under Grant DE-FG02-05ER25710 and by LLNL under Contract DE-AC5207NA27344 under the auspices of the US Department of Energy Advanced Scientific Computing Research's Multiscale Mathematics Initiative.



## Appendix A. Numerical Regularization

A number of numerical and modeling poles must be circumnavigated to implement the MLMC method successfully.

### *Diffusion to negative speeds*

The Langevin equation governing the evolution of  $v$  is (50). Finite changes in  $\Delta v$  arising from terms containing  $\Delta W_v$  can be of any size, although large values are (exponentially) unlikely. It follows that  $v(t + \Delta t_l) = v(t) + \Delta v$  can be such that  $v(t + \Delta t_l) \leq 0$ . This is not only unphysical, but also numerically problematic as  $F, D_a$  become singular at a rate  $v^{-1}, v^{-2}$  respectively as  $v \rightarrow 0^+$ . Furthermore, the deterministic drag coefficient  $F$  is anti-symmetric in  $v$ , so if  $v < 0$  the first term in (50) drives  $v$  to yet more negative values.

Our approach is to regularize the coefficient of the deterministic drag term  $F$  in (50), and the stochastic diffusion term  $D_a$  in (51). Our method differs from that of Lemons et al. [27] who did not account for the small, but finite, probability case that particles diffuse to  $v < 0$ , even when the coefficients of the diffusion terms are set to zero for small  $v$ . We define the piecewise Lipschitz continuous functions

$$\mathcal{F}(v) = \begin{cases} F(v), & v > v_c \\ \frac{F'(v_c)}{2v_c}(v^2 - v_c^2) + F(v_c), & v \leq v_c \end{cases}, \quad (\text{A.1})$$

and

$$\mathcal{D}_a(v) = \begin{cases} D_a(v), & v > v_c \\ \frac{D'_a(v_c)}{2v_c}(v^2 - v_c^2) + D_a(v_c), & v \leq v_c \end{cases}, \quad (\text{A.2})$$

where  $v_c$  is the critical value of  $v$  at which regularization occurs, and  $F', D'_a = dF/dv, dD_a/dv$ .

Direct substitution of (A.1) into (50) yields a regularized equation for  $\Delta v$ :

$$\Delta v = \mathcal{F} \Delta t_l + \sqrt{2D_v} \Delta W_v + \kappa_M D'_v \frac{1}{2} (\Delta W_v^2 - \Delta t_l). \quad (\text{A.3})$$

An analogous modification of (51) also follows, but further regularization of the imaginary diffusion coefficients is first required. We note that in the small region  $v < v_c$ , the Einstein relation (56) is not obeyed. In the simulations conducted here we set  $v_c = 0.05$ , which we find, empirically, to work.

### *Imaginary coefficients*

The Langevin equation governing the evolution of  $\mu$  is (51). Analogous to the previous section, finite changes in  $\Delta \mu$  driven by large values of  $\Delta W_\mu, A_{v\mu}$  can result in  $\mu(t + \Delta t_l) = \mu(t) + \Delta \mu$  being such that  $|\mu| > 1$ . It follows that  $\sqrt{1 - \mu^2}$  can become imaginary, which is unphysical.

To constrain the discretized equations to be physical, we define the modified coefficient  $\mathcal{M}$  to be:

$$\mathcal{M}(\mu) = \begin{cases} \sqrt{1 - \mu^2} & |\mu| < \mu_c \\ \sqrt{1 - \mu_c^2} \exp[(\mu - \mu_c)S(\mu_c)] & |\mu| \geq \mu_c \end{cases}, \quad (\text{A.4})$$

where  $S(\mu_c) = \mu_c/(1 - \mu_c^2)$ , and  $\mu_c$  is the critical value at which regularization occurs.

The coefficient is unaltered away from the critical poles, and regularized near them when  $|\mu| > \mu_c$  in a manner consistent with the Einstein relation (55) and the condition that the coefficients are Lipschitz continuity.

Substituting (A.2) and (A.4) into (51), the regularized evolution equation for  $\Delta\mu$  is

$$\Delta\mu = 2\mathcal{D}_a \mathcal{M} \mathcal{M}' \Delta t_l + \sqrt{2\mathcal{D}_a \mathcal{M}} \Delta W_\mu + \kappa_M \left[ \mathcal{D}_a \mathcal{M} \mathcal{M}' (\Delta W_\mu^2 - \Delta t_l) + \sqrt{\frac{D_v}{\mathcal{D}_a}} \mathcal{M} \mathcal{D}_a' A_{v\mu} \right], \quad (\text{A.5})$$

where  $\mathcal{M}' = d\mathcal{M}/d\mu$ . In the simulations conducted here we set  $\mu_c = 0.95$  which, again, we find to work empirically.

## References

- [1] S. Koh, C. Chang, S. Ku, J. Menard, H. Weitzner, W. Choe, Bootstrap current for the edge pedestal plasma in a diverted tokamak geometry, *Physics of Plasmas* 19 (2012) 72505.
- [2] G. Park, C. Chang, I. Joseph, R. Moyer, Plasma transport in stochastic magnetic field caused by vacuum resonant magnetic perturbations at diverted tokamak edge, *Physics of Plasmas* 17 (2010) 102503.
- [3] B. Cohen, L. Divol, A. Langdon, E. Williams, Effects of ion-ion collisions and inhomogeneity in two-dimensional kinetic ion simulations of stimulated Brillouin backscattering, *Physics of plasmas* 13 (2006) 022705.
- [4] R. Bosch, R. Berger, B. Failor, N. Delamater, G. Charatis, R. Kauffman, Collision and interpenetration of plasmas created by laser-illuminated disks, *Physics of Fluids B: Plasma Physics* 4 (1992) 979–988.
- [5] A. A. Schekochihin, S. C. Cowley, W. Dorland, G. W. Hammett, G. G. Howes, E. Quataert, T. Tatsuno, Astrophysical Gyrokinetics: Kinetic and Fluid Turbulent Cascades in Magnetized Weakly Collisional Plasmas, *ApJ* 182 (2009) 310–377.
- [6] T. Tatsuno, W. Dorland, A. Schekochihin, G. Plunk, M. Barnes, S. Cowley, G. Howes, Nonlinear phase mixing and phase-space cascade of entropy in gyrokinetic plasma turbulence, *Physical review letters* 103 (2009) 15003.
- [7] M. Giles, Multilevel Monte Carlo path simulation, *Operations Research* 56 (2008) 607–617.
- [8] M. Giles, Multilevel Monte Carlo methods, *Monte Carlo and Quasi-Monte Carlo Methods 2012* (2013).
- [9] H.-J. Bungartz, M. Griebel, Sparse grids, *Acta numerica* 13 (2004) 147–269.
- [10] L. Landau, Kinetic equation for the Coulomb effect, *Phys. Z. Sowjetunion* 10 (1936) 154.
- [11] R. E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, *Acta numerica* 1998 (1998) 1–49.
- [12] A. Thomas, M. Tzoufras, A. Robinson, R. Kingham, C. Ridgers, M. Sherlock, A. Bell, A review of Vlasov–Fokker–Planck numerical modeling of inertial confinement fusion plasma, *Journal of Computational Physics* 231 (2012) 1051–1079.
- [13] L. Ricketson, M. Rosin, R. Caflisch, A. Dimits, An entropy based thermalization scheme for hybrid simulations of Coulomb collisions, *arXiv preprint arXiv:1301.5678* (2013).
- [14] T. Takizuka, H. Abe, A binary collision model for plasma simulation with a particle code, *Journal of computational physics* 25 (1977) 205–219.
- [15] K. Nanbu, Theory of cumulative small-angle collisions in plasmas, *Physical Review E* 55 (1997) 4642–4652.
- [16] A. V. Bobylev, I. Potapenko, Monte Carlo methods and their analysis for Coulomb collisions in multicomponent plasmas, *Journal of Computational Physics* 246 (2013) 123.
- [17] A. Bobylev, K. Nanbu, Theory of collision algorithms for gases and plasmas based on the Boltzmann equation and the Landau-Fokker-Planck equation, *Physical Review E* 61 (2000) 4576–4586.
- [18] C. Wang, T. Lin, R. Caflisch, B. Cohen, A. Dimits, Particle simulation of Coulomb collisions: Comparing the methods of Takizuka & Abe and Nanbu, *Journal of Computational Physics* 227 (2008) 4308–4329.
- [19] G. Dimarco, R. Caflisch, L. Pareschi, Direct simulation Monte Carlo schemes for Coulomb interactions in plasmas, *arXiv preprint arXiv:1010.0108* (2010).
- [20] M. Ivanov, V. Shvets, The use of the particle method to simulate a collisional plasma, in: *Akademiia Nauk SSSR Doklady*, volume 238, 1978, pp. 1324–1327.
- [21] M. Ivanov, V. Shvets, The method of stochastic differential equations for computing the kinetics of a plasma with collisions, *USSR Computational Mathematics and Mathematical Physics* 20 (1980) 146–155.
- [22] S. Painter, Data-parallel algorithms for Monte Carlo simulation of neoclassical transport in magnetically confined plasmas, *Computer physics communications* 77 (1993) 342–356.
- [23] M. E. Jones, D. S. Lemons, R. J. Mason, V. A. Thomas, D. Winske, A grid-based Coulomb collision model for PIC codes, *Journal of Computational Physics* 123 (1996) 169–181.

- [24] W. M. Manheimer, M. Lampe, G. Joyce, Langevin representation of Coulomb collisions in PIC simulations, *Journal of Computational Physics* 138 (1997) 563–584.
- [25] S. Dettrick, H. Gardner, S. Painter, Monte Carlo transport simulation techniques for stellarator fusion experiments, *Australian journal of physics* 52 (1999) 715–732.
- [26] M. Sherlock, A Monte-Carlo method for Coulomb collisions in hybrid plasma models, *Journal of Computational Physics* 227 (2008) 2286–2292.
- [27] D. Lemons, D. Winske, W. Daughton, B. Albright, Small-angle Coulomb collision model for particle-in-cell simulations, *Journal of Computational Physics* 228 (2009) 1391–1403.
- [28] B. I. Cohen, A. M. Dimits, A. Friedman, R. E. Caflisch, Time-step considerations in particle simulation algorithms for Coulomb collisions in plasmas, *Plasma Science, IEEE Transactions on* 38 (2010) 2394–2406.
- [29] J. Qiang, R. D. Ryne, S. Habib, Self-consistent Langevin simulation of Coulomb collisions in charged-particle beams, in: *Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)*, IEEE Computer Society, 2000, p. 27.
- [30] J. Velasco, A. Bustos, F. Castejón, L. Fernández, V. Martin-Mayor, A. Tarancón, Isdep: Integrator of stochastic differential equations for plasmas, *Computer Physics Communications* (2012).
- [31] M. Cadjan, M. Ivanov, Stochastic approach to laser plasma kinetics with Coulomb collisions, *Physics Letters A* 236 (1997) 227–231.
- [32] M. Cadjan, M. Ivanov, Langevin approach to plasma kinetics with Coulomb collisions, *Journal of plasma physics* 61 (1999) 89–106.
- [33] A. Dimits, B. Cohen, R. Caflisch, M. Rosin, L. Ricketson, Higher-order time integration of Coulomb collisions in a plasma using Langevin equations, *Journal of Computational Physics* 242 (2013) 561–580.
- [34] M. Giles, Improved multilevel Monte Carlo convergence using the Milstein scheme, *Monte Carlo and quasi-Monte Carlo methods 2006* (2008) 343–358.
- [35] M. N. Rosenbluth, W. M. MacDonald, D. L. Judd, Fokker-Planck equation for an inverse-square force, *Physical Review* 107 (1957) 1.
- [36] N. G. Van Kampen, *Stochastic processes in physics and chemistry*, volume 1, North holland, 1992.
- [37] P. E. Kloeden, E. Platen, *Numerical solution of stochastic differential equations*, volume 23, Springer Verlag, 1992.
- [38] A. Einstein, Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen, *Annalen der physik* 322 (1905) 549–560.
- [39] M. B. Giles, L. Szpruch, Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation, *arXiv preprint arXiv:1202.6283* (2012).
- [40] P. Lévy, Wiener’s random function, and other Laplacian random functions, in: *Second Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 1951, pp. 171–187.
- [41] J. G. Gaines, T. J. Lyons, Variable step size control in the numerical solution of stochastic differential equations, *SIAM Journal on Applied Mathematics* 57 (1997) 1455–1484.
- [42] J. G. Gaines, T. J. Lyons, Random generation of stochastic area integrals, *SIAM Journal on Applied Mathematics* 54 (1994) 1132–1146.
- [43] F. Mezzadri, How to generate random matrices from the classical compact groups, *Notices Amer. Math. Soc.* 54 (2007) 592–604.
- [44] B. I. Cohen, A. M. Dimits, D. J. Strozzi, A grid-based binary model for Coulomb collisions in plasmas, *Journal of Computational Physics* 234 (2013) 33–43.
- [45] B. Trubnikov, *Review of plasma physics* 1, 1965.
- [46] H. Hoel, E. Von Schwerin, A. Szepessy, R. Tempone, Adaptive multilevel Monte Carlo simulation, in: *Numerical Analysis of Multiscale Computations*, Springer, 2012, pp. 217–234.
- [47] M. B. Giles, B. J. Waterhouse, Multilevel quasi-Monte Carlo path simulation, *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics* (2009) 165–181.
- [48] T. Gerstner, M. Noll, Randomized multilevel quasi-Monte Carlo path simulation, 2013.
- [49] L. Ricketson, Three improvements to multi-level Monte Carlo simulation of SDE systems, *arXiv preprint arXiv:1309.1922* (2013).
- [50] L. Szpruch, M. B. Giles, Antithetic multilevel Monte Carlo estimation for multidimensional SDEs, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, Springer (2013) 297–312.
- [51] E. Hirvijoki, A. Brizard, A. Snicker, T. Kurki-Suonio, Monte Carlo implementation of a guiding-center Fokker-Planck kinetic equation, *arXiv preprint arXiv:1305.2013* (2013).
- [52] M. B. Giles, D. J. Higham, X. Mao, Analysing multi-level Monte Carlo for options with non-globally Lipschitz payoff, *Finance and Stochastics* 13 (2009) 403–413.
- [53] C. Cercignani, *The Boltzmann Equation and Its Applications*, number 67 in *Applied Mathematical Sciences*, Springer-Verlag, 1988.
- [54] G. W. Hammett, F. W. Perkins, Fluid moment models for Landau damping with application to the ion-temperature-gradient instability, *Physical review letters* 64 (1990) 3019–3022.

- [55] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, *Journal of Computational Physics* 149 (1999) 201–220.